SMILE User's Manual

Table of Content

1. Introduction

The Sparse Multidimensional Iterative Lineshape Enhanced (SMILE) algorithm integrates *a priori* information about NMR signals for most robust reconstruction of non-uniformly sampled (NUS) multidimensional data. It also treats the data as one single spectrum instead of many cross sections, further increasing robustness at the expense of an increase in the computational burden. The algorithm has been implemented as a new NMRPipe function, called SMILE. Together with such new tools as nusExpand.tcl and nusZF.com recently released by Frank Delaglio, the SMILE processing function streamlines the NUS spectral reconstruction. This manual is written to help users in the NMRPipe community set up a multithreading SMILE reconstruction job on a multicore Linux workstation or on a Mac computer.

2. How to obtain the program

The SMILE processing function is available as a plug-in for the NMRPipe package released by Frank Delaglio on November 24, 2015. Users are required to install this latest version of NMRPipe, which can be downloaded from:

http://spin.niddk.nih.gov/NMRPipe/install/

Note that File 4 (plugin.smile.tZ) must be downloaded in order for SMILE to run, although it is not a required component for NMRPipe. This plug-in file includes the actual executable program (nusPipe) that provides the SMILE processing function. Depending on the system on which NMRPipe is installed, nusPipe placed in the nmrbin.linux9, nmrbin.linux212_64, or nmrbin.mac folder under the $NMRBASE directory is called during the processing. In addition, the following three environmental variables are set in the nmrInit.mac.com, nmrInit.linux9.com, or nmrInit.linux212_64.com initialization script:

```
setenv NMR_PLUGIN_FN   SMILE
setenv NMR_PLUGIN_EXE  nusPipe
setenv NMR_PLUGIN_INFO "[-nDim -sample sName ...] MD NUS Reconstruction"
```

Future developments and updates of SMILE will be posted on this site. Users then need to simply download the updated executables and replace those in the current release.

To test if SMILE is properly installed on a computer, one can enter `"nmrPipe -fn SMILE -help"` on a terminal. If the SMILE plug-in is correctly set up, it should print the following help page on the screen:

```
*-----------------------------------------------------------------*
 | This is the Beta version of SMILE processing function. Please  |
 | contact Jinfa Ying <jinfaying@niddk.nih.gov> for any bugs you  |
 | may find. Your comments/suggestions are much appreciated too!  |
 | Also reference to Ying, J. et al., JBNMR, in preparation.      |
 *-----------------------------------------------------------------*

SMILE: Sparse Multidimensional Iterative Lineshape Enhanced.
 -nDim    [4]      Number of Dimensions.
 -[x]zf   [2]      Zero Fill (Power of 2);
                   For indirect time-domains only.
                   (use -yzf -zzf etc, or for all indirect
                   dimensions when no axis is specified.)
 -[x]zfSize[0]     Zero Fill size, larger than -T value in fid.com;
                   For indirect time-domains only.
                   (use -yzfSize -zzfSize etc
                   If provided, -zf ignored
 -[x]Alt           Use sign alternation for FT;
                   For indirect time-domains only.
                   (use -yAlt -zAlt etc, or for all indirect
                   dimensions when no axis is specified.)
 -[x]Neg           Negate imaginary for FT;
                   For indirect time-domains only.
                   (use -yNeg -zNeg etc, or for all indirect
                   dimensions when no axis is specified.)
Phasing Parameters (also use -y.. -z.. etc);
 -xP0     [0]      Zero Order Phase Correction;
 -xP1     [0]      First Order Phase Correction;
Apodization Parameters (also use -y.. -z.. etc);
applies to all indirect dimensions if no axis is specified;
defines window used on SMILE data:
 -[x]Apod fName    Window Function File Name.
 -[x]Apod [SP]     Window Function: EM GM SP JMOD.
 -[x]Q1   [0.50]   Window Function Parameter Q1.
 -[x]Q2   [0.98]   Window Function Parameter Q2.
 -[x]Q3   [1.00]   Window Function Parameter Q3.
Three additional parameters for the Sine window only;
   and are ignored when -Apod is not SP;
 -[x]ELB  [0.0]    Exponential, Hz. (LB)
 -[x]GLB  [0.0]    Gaussian, Hz.    (GB)
 -[x]GOFF [0.0]    Gauss Offset, 0 to 1. (GOFF)
SMILE Parameters:
 -sample sName     NUS sampling schedule.
 -sampleCount sCount No. of valid samples in the schedule.
 -off     [0,0,0]  Offset for the sampling schedule in each indirect dim.
 -nThread [1]      Number of threads to be used.
 -thresh  [0.80]   Threshold for peak picking.
 -subConst[1.00]   Peak intensity downscaling factor before being subtracted.
 -maxNPks [0]      Max # of peaks per Slice/Plane/Cube for 2/3/4D in each iter.
                   Zero for SMILE to automatically determine.
                   For 2D, SMILE always automatically set to 1.
 -nSimPks [30]     No. of simulated peaks for LW correction.
 -minTDL  [0.1]    Min TD Length (AcqT/T2) to be simulated.
 -maxTDL  [10.0]   Max TD Length (AcqT/T2) to be simulated.
 -maxMem  [0.0]    Maximum memory in GBytes.
Constant/Mixed Time acquisition (also use -y and -z...);
 -xCT     [0]      Constant Time flag.
                   =0 for Real time evolution.
                   =td/2 for CT time evolution.
                   <td/2 for Mixed time evolution.
Convergence Tests:
 -maxIter [200]    Maximum Iteration Count.
 -sigma   [0.0]    RMS noise of the spectrum;
                   Zero for Auto Estimates.
 -nSigma  [6]      No. of sigma to be considered as peaks.
Notes:
 1. Important options are -nDim -sample -nThread
    -maxIter -[x/y/z]zf -[x/y/z]P0/P1
    -[x/y/z]Apod/Q1/Q2/Q3
 2. Underestimate -sigma * -nSigma
    --> pick up noises as peaks.
 3. Overestimate -sigma * -nSigma
    --> miss peaks.
 4. If possible, use 2-fold zero fills.
```

3. Hardware and OS requirements

Currently, the SMILE function is only ported for Linux and Mac OS X. Both 32-bit linux9 and 64-bit linux212_64 versions were compiled on CentOS 6.7 using the Intel compiler icc (15.0.1.133/20141023) and the glibc2.12 library (newer than glibc2.8 used for the linux9 version of the main NMRPipe package, see the installation site above). This means that linux9 NMRPipe users may be able to call the other processing functions, but not SMILE, unless glibc2.12 or above is available. The Intel libraries required by both Linux versions of SMILE are statically linked such that users without those installed libraries can still run the program. The Mac version was built on OS X 10.6.8 (also newer than 10.5.8 used for the mac version of NMRPipe) using gcc 4.9, and includes both the i386 and x86_64 architectures, but not the PowerPC architecture. All SMILE-builds allow multithreading spectral reconstruction on a shared memory computer system via the OpenMP API. Although not required, the OpenMP environmental variables (staring with OMP_) and the GNU extensions (for the mac version, starting with GOMP_) as well as the Intel extensions (for the Linux versions, starting with KMP_) may be used to optimize the SMILE calculation. Note that OMP_DYNAMIC and OMP_SCHEDULE are set internally by SMILE, and OMP_NUM_THREADS is set through the "-nThread" option described below. Setting these environmental variables has no effect on a SMILE job. For more details about the other environmental variables, visit the following or any other OpenMP websites:

https://computing.llnl.gov/tutorials/openMP/#EnvironmentVariables
https://gcc.gnu.org/onlinedocs/libgomp/Environment-Variables.html#Environment-Variables
https://software.intel.com/en-us/node/522775

Depending on the size of the expanded NUS data, SMILE may demand a large amount of memory for a 3D or 4D reconstruction. Users can estimate the minimal amount of memory SMILE needs from the size of the data matrix after the direct dimension is processed. For the 4D reconstruction example discussed below (Section 6.5), after the direct dimension (t_4) is processed, the data matrix consists of 56* (t1, complex) × 80* (t2, complex) × 80* (t3, complex) × 646 (F4, real) points. The total file size of the input data into SMILE then is 1.85×10^9 data points, or 6.9 GB. With at least two fold zero fills in all the indirect dimensions for FT (resulting in 256* × 320* × 320* × 646 for the spectrum), SMILE prefers approximately 520 GB of memory, but can run fine with a minimum of ~17 GB memory by setting the -maxMem option accordingly (see Section 5.5), i.e. roughly only 2.5 times the size of the input files. This factor may be dropped further to 1.5 in future releases. If SMILE is started on a system without sufficient memory, an error is printed and the program quits.

4. General considerations and procedures

To make NUS reconstruction as close to the conventional data processing as possible, SMILE requires the non-uniformly sampled Bruker or Varian data to be

sorted and expanded, with the unsampled points filled with zeros. Although this can be done using the nusExpand.tcl script either before or after the time-domain data is converted to the NMRPipe format, doing this prior to the data conversion is advantageous because the expanded data can be converted by a conventional script without any unusual changes. For example, if any indirect dimension is acquired with the Echo-AntiEcho quadrature mode, the time-domain data reshuffling can be done directly during the conversion. If the conversion is done before the expansion, the data must be first treated in a complex mode, and then be sorted and expanded, followed by an NMRPipe macro to extract the real and imaginary components from each Echo-AntiEcho pair. One disadvantage of expanding the data first is the generation of a very large time-domain data file. However, the file can be deleted once the data is converted. Another disadvantage is that this approach does not work for any data acquired in the magnitude mode (QF flag on Bruker). However, as SMILE is intended for phase-sensitive experiments, it should not be a problem to always run nusExpand.tcl before the data conversion, although the other approach can be used too.

Because SMILE begins from the data in the NMRPipe format, it works equally well for Bruker, Varian or any other type of raw NUS data, provided they are correctly converted prior to the reconstruction. Although all examples given in this manual are collected on Bruker spectrometers, Varian users can set up and run SMILE in the same way. For the details about setting up a conversion script for Varian NUS data, refer to the help page of the "varian" script in the NMRPipe package by entering "varian -help" on a terminal.

Once the data is expanded and converted, the direct dimension must be first apodized, zero filled, Fourier transformed, and phased and the imaginary must be discarded. To minimize the size of the data matrix, the spectral region of interest should be extracted in the direct dimension. In addition, it is required that the direct dimension (along the x-axis in NMRPipe) is transposed to the last dimension (i.e. y-, z-, or a-axis for 2D, 3D, or 4D), while the relative order of the other indirect dimensions should not be altered during the transposition. This means that for a 4D data set, the indirect dimensions originally along the y-, z-, and a-axis must be transposed to the x-, y-, and z-axis, respectively, and the direct dimension must be stored along the a-axis. See the 3D and 4D examples (Sections 6.3-6.5) for how this is done. SMILE also requires the sampling schedule to be arranged accordingly, i.e. with the first column corresponding to the x-axis, second column to the y-axis, and the third column to the z-axis for the 4D. No axis reordering can be done at the runtime.

Similar to other NUS reconstruction algorithms, the quality of the SMILE reconstruction may deteriorate if the baseline in the direct dimension is poorly distorted. Users are advised to collect the data with care to keep the baseline as flat as possible, although this may be difficult in certain types of experiments. For Bruker users, the oversampling should be handled during the processing of the direct dimension instead of the conversion step (as done for all examples shown in

this manual), which should help improve the baseline flatness. If the spectral region of interest in the direct dimension is narrow (often the case for intrinsically disordered proteins) but the data is collected with a reasonably large width, users may have a sufficiently large baseline segments for making a reasonable correction during the processing of the direct dimension at the expense of a very slightly elevated noise level. For the indirect dimensions, it is highly recommended that the first order phase correction should be as close to 0°, 180° or 360° as possible, although a small deviation from these values may be well tolerated during the reconstruction. For a given indirect dimension acquired with a first order phase of 360°, the time-shifting property of FT can be employed to easily eliminate the linear phase correction and to avoid the constant baseline offset. An example of this is shown below for a 3D HNCO reconstruction (Section 6.3).

Although not extensively tested, SMILE should work for reconstructing a spectrum with strong axial peaks on the edge(s) of one or more indirect dimensions. However, the presence of strong axial peaks slows down the SMILE reconstruction significantly. Collecting data with no or minimal axial peaks generally will improve the performance. In addition, NMR data with fairly strong t_1 noise ridges (e.g. from the residual water peak in the ^{13}C noesy-hsqc example in Section 6.4 below) can be reconstructed too, although the calculation may run more slowly or may take more iterations to complete.

SMILE stops if it either reaches the maximum number of iterations specified by the "-maxIter" option or detects no more peaks above a threshold that is based on the values set by the "-sigma" or "-nSigma" options. Before SMILE exits, it outputs the reconstructed time-domain data onto the unix pipe such that a conventional NMRPipe processing can continue or the reconstructed data can be first saved by the pipe2xyz program. The SMILE time-domain output data have the same size, format and axis order as the input data, except that the zeros for the unsampled points are now replaced with the reconstructed values. Conventional NMRPipe processing can then be used to obtain the final spectrum. Note that the indirect dimensions of the reconstructed data remain apodized, and no window function should be applied again unless one wants to use a different window during the conventional processing, in which case the user can use the corresponding NMRPipe function with the "-hdr -inv" option to invert the window and then apply a new apodization function before the FT of the indirect dimension(s).

For best reconstruction, it is recommended that users first treat the expanded data as being fully sampled and process it without a SMILE reconstruction. By monitoring some of the strongest peaks, this allows users to examine the phase correction in all dimensions, to determine an optimal spectral region in the direct dimension, and to decide if the imaginary of a particular indirect dimension needs to be negated or if the sign in an indirect dimension needs to be alternated. If the conventional NMRPipe FT function requires –alt or -neg to get a correct spectrum,

then a similar flag is required for that dimension during the SMILE reconstruction. See the -[x/y/z]Alt or -[x/y/z]Neg SMILE options (Section 5.12) below.

5. SMILE command line options

After processing of the direct (fully sampled) dimension, the SMILE function can be called for reconstruction of the unsampled data. A SMILE reconstruction can be fine-adjusted by setting many command line arguments, although most of these already have reasonable default values and therefore are not required for most reconstructions. For example, for the 3D HNCO data discussed in Section 6.3, only 8 options need to be explicitly defined for the reconstruction:

```
xyz2pipe -in ft1/test%04d.ft1 -x                          \
| nmrPipe  -fn SMILE -nDim 3 -sample nuslist -nThread 32   \
           -sampleCount 800 -nSigma 5 -off 0 -1 -report 1  \
           -xCT 43                                         \
| pipe2xyz -out ft1/rc%04d.ft1 -x
```

This helps keep the SMILE command line short and concise, as compared to the following fully specified but lengthy line for doing exactly the same reconstruction:

```
xyz2pipe -in ft1/test%04d.ft1 -x                          \
| nmrPipe  -fn SMILE -nDim 3 -sample nuslist -nThread 32   \
           -sampleCount 800 -nSigma 5 -off 0 -1 -report 1  \
           -maxIter 200 -maxMem 0 -thresh 0.8 -sigma 0     \
           -xCT 43 -xApod SP -xQ1 0.5 -xQ2 0.98 -xQ3 1     \
           -yCT  0 -yApod SP -yQ1 0.5 -yQ2 0.98 -yQ3 1     \
           -xzf 2 -xP0 0 -xP1 0                            \
           -yzf 2 -yP0 0 -yP1 0                            \
| pipe2xyz -out ft1/rc%04d.ft1 -x
```

5.1 -nDim

The first option "-nDim" tells SMILE the total number of dimensions, including the directly detected dimension, which is required for SMILE to properly initialize the array space. SMILE works for 2D, 3D or 4D. If the "-nDim" is outside the range of 2-4, an error is printed and the program quits. The default value is 4.

5.2 -sample and -sampleCount

The "-sample" option allows users to provide SMILE with the NUS sampling schedule. If SMILE fails to read the list or if no list is provided, it aborts with an error message. While reading the list, SMILE ensures the sampling lists are consistent with the sizes of the indirect dimensions, and otherwise stops with a specific error displayed. One optional command line argument closely related to "-sample" is "-sampleCount". The default count is the number of entries (pairs for 3D and triples for 4D) in the entire sampling list. However, if one stops the NUS experiment before it finishes, the "-sampleCount" option allows the user to use the original list without it being truncated. Only the first -sampleCount entries in the list are then read by SMILE. This option also can be useful for further down-sampling a NUS data set, for example, to examine how the reconstructed spectral

quality is impacted by the amount of sampling. This works because any data points not in the sampling list read by SMILE are automatically set to zeros, even if they are experimentally sampled.

5.3 -x/y/zP0 and –x/y/zP1

For optimal reconstruction, SMILE needs to know the required zero order (P0) and first order (P1) phase corrections in each of the indirect dimensions. An axis name must precede P0 and P1, i.e. the option -P0 or -P1 is not valid, but -xP0 or -xP1 is. As mentioned in Section 5.7 for the apodization options, the -x/y/zP1 value is used for determining the first time point scaling factor for the x-, y-, and z-dimensions, respectively. The scaling factor is 0.5 if the first order phase correction is in the range of (-45°, +45°), and otherwise set to 1.0. The first point in each indirect dimension remains scaled in the final reconstructed data.

5.4 -nThread

Unless SMILE is used to reconstruct a 2D spectrum, the "-nThread" option (default: 1) should be specified. This allows SMILE to use multiple CPU cores to run some of the calculations in parallel via OpenMP multithreading. The number of threads one should use depends on the size of the data, but in general no more threads than the number of CPU cores should be used.

5.5 -maxMem

As mentioned above, the "-maxMem" option is useful when users do not have a sufficient amount of computer memory for a large data reconstruction. The default value is zero and SMILE tries to allocate as much memory as it needs. If the memory is insufficient, SMILE aborts with an error message. This can be avoided if the system has a memory larger than the minimal amount of space required by SMILE (ca 2.5 times the size of the converted input file, after the direct dimension is processed and the desired region is extracted). The -maxMem restricts the amount of memory SMILE can use and makes the program automatically employ a memory saving mode. However, it may still quit if less than the specified amount of memory is actually available or if the -maxMem amount is smaller than the required minimum.

5.6 -[x/y/z]zf and -[x/y/z]zfSize

The total amount of memory required for running the SMILE function strongly depends on the amount of zero filling in all the dimensions. Although a double zero fill for each dimension is desirable (i.e., 4 times the original time domain length), it increases the data size and potentially can make the reconstruction intractable due to a too slow reconstruction speed or an insufficient amount of memory. A one-fold zero fill may suffice, provided that the digital resolution (i.e. Hz/point) after the zero-filling and FT is ca 35% of the full line width (Hz) at the half height or better.

The amount of zero filling in each indirect dimension is controlled by -xzf, -yzf, and -zzf (or just -zf simultaneously for all the indirect dimensions) and the default value for those arguments is 2. The size after the zero-fill always automatically rounds up to the next power of 2. For example, if the size in an indirect dimension is 56* complex pairs, a two-fold zero fill followed by the rounding gives 256*. Likewise, a size of 80* complex pairs with the same zero filling and rounding yields 512*. To suppress the automatically rounding, SMILE allows users to provide the actual size for an indirect dimension to be zero filled to, using the options of -xzfSize, -yzfSize, or -zzfSize. This way, a user can zero fill 80* to 320* instead of 512*. The FT library used by SMILE can transform a vector of any length and the performance is not adversely impacted even if the size is not a power of 2.

5.7 -[x/y/z]Apod, -[x/y/z]Q1, -[x/y/z]Q2, -[x/y/z]Q3

During the SMILE processing, all the indirect dimensions are apodized. The "-Apod" specifies the type of window function and can take SP, EM, GM, TM, TRI, GMB, and JMOD for sine, exponential, Gaussian, trapezoid, triangle, another version of Gaussian, and J modulation profile, respectively. These functions are identical to the standard NMRPipe function of the same name. The additional "-Q1", "-Q2", and "-Q3" options provide up to 3 parameters for completely defining each function. For more detailed information, refer to the following NMRPipe webpage: <http://spin.niddk.nih.gov/NMRPipe/ref/nmrpipe/apod.html>. For SP, the "-Q1" corresponds to "-off", "-Q2" to "-end", and "-Q3" to "-pow". The "-Q3" option may be ignored for those functions that require only 2 parameters. For example, when "-Apod" is set to GMB, "-Q1" is used for "-lb" and "-Q2" for "-gb", while "-Q3" is not used. The default window functions in SMILE is SP with "-Q1" set to 0.5, "-Q2" to 0.98, and "-Q3" to 1.

An axis designation may precede these options for applying a window function to a particular dimension. For instance, "-xApod GM –xQ1 -5 –xQ2 3, -xQ3 0.5" set the first indirect dimension to use a Gaussian window. The options with an axis specified take precedence over those containing no axis. For example, for a 4D data set, "-Apod SP -xApod GM" applies a Gaussian function to the x-dimension, and a sine window to the y- and z-dimensions. In the 3D ^{13}C NOESY-CT-HSQC example (Section 6.4 below), we show how the GM and SP windows are applied to the x- and y-dimensions, respectively.

The first time point scaling is determined by SMILE based on the first order phase correction. It is automatically set to 1.0 or 0.5 if the -xP1, -yP1, or -zP1 is in the range from -45° to +45°. No SMILE option is available for setting the scaling factor to a value other than 0.5 or 1.0.

5.8 -thresh

This option defines the threshold of the weakest peaks relative to the most intense peak to be reconstructed in each iteration. The default value is 0.8. For a 2D

reconstruction, the option becomes irrelevant because only the strongest peak is selected. For faster 3D and 4D reconstruction, this may even dropped to 0.5, while a slightly higher value than the default can sometimes improve the performance at the expense of a potentially longer computation time. For example, for the 3D ^{13}C NOESY-HSQC spectrum in Section 6.4 (high number of peaks and large dynamic range), setting "-thresh" to 0.95 leads to a slightly better reconstruction.

5.9 -sigma and -nSigma

Those two options allow users to provide the intrinsic thermal noise level and to define the weakest peaks that can be reconstructed. The default value for -sigma is 0, requiring SMILE to automatically determine the intrinsic noise level. The default for -nSigma is 6. This works reasonably well for 4D, although a slightly lower value (4 or 5) may improve the performance for 2D and moderate size 3D spectra.

5.10 -maxIter

This option defines the maximum number of iterations. SMILE stops the calculation and outputs the reconstructed data, even if noise threshold has not yet been reached. The default value is 200. For fast trial reconstructions, users may want to use a low -maxIter value in combination with a lowered -thresh value, discussed in Section 5.8. However, for the final reconstruction, a high number of iterations is recommended for optimal SMILE performance to ensure that the iterative process doesn't truncate prematurely.

5.11 -x/y/zCT

This option tells SMILE if any indirect dimension is acquired in a constant-time or mixed-time manner. The default value is zero, which indicates a dimension is recorded by real-time increments. If the x-dimension is recorded in CT, then -xCT should be set to the number of complex pairs (i.e. the same as -xT in the fid.com conversion script). However, if the x-dimension is acquired in mixed time (i.e. first as a constant-time and then followed by real-time increments), then -xCT should be set to the last increment before the real-time increment starts. For example, in the 3D HNCO example discussed in Section 6.3, the ^{15}N dimension has 200* complex pairs for a total acquisition time of 108.8 ms. This exceeds the typical 25ms $^{1}J_{NC'}$ rephasing time. As a result, the first 44* pairs (~23.4 ms) is recorded in constant time, while the rest in real-time. In this case, -xCT should be set to 43 because the index is zero-based. Users can determine this value by first finding the initial length of the delay to be decremented and the actual decrement. In the HNCO experiment, d0 is decremented and has an initial value of 11.74 ms. The decrement for d0 (i.e. in0) is 0.272 ms. -xCT is therefore set to the integer obtained by rounding down the ratio of 11.74/0.272 = 43.2.

Currently, the -x/y/zCT value cannot be set to be larger than the corresponding -x/y/zT values in the fid.com conversion script. Otherwise, the reconstruction exits with an error at some point.

The CT option can help ensure optimal reconstruction, but its use is not very critical as long as the indirect dimension is properly apodized. Users therefore may choose to ignore it.

5.12 -[x/y/z]Neg and -[x/y/z]Alt

These two options are equivalent to the -neg and -alt options of the conventional FT function in the NMRPipe package. Enter "nmrPipe -fn FT -help" for additional information. If "nmrPipe -fn FT" of the x-, y-, or z-dimension requires -neg or -alt to get a correct spectrum, then –x/y/zNeg and –x/y/zAlt are also required for SMILE. When no axis is specified, the options apply to all the indirect dimensions. It must be pointed out that the -Neg option is not critical at all because it only reverses the spectrum during the SMILE reconstruction, and data can still be correctly reconstructed even without being reversed. However, the -Alt is absolutely required by SMILE if FT needs –alt and the linear phase correction in the corresponding dimension is not zero. The NMRPipe FT function needs -alt and SMILE requires -Alt if one acquires an indirect dimension using Bruker's standard States-TPPI mode. To verify those settings, users are recommended to carry out a "quick and dirty" NMRPipe processing of the expanded NUS data without SMILE reconstruction or by setting the -maxIter option to 0.

5.13 -off

This is an option that allows users to provide the offset in their sampling schedule for each indirect dimension. It serves the same purpose as the "-off" option in the nusExpand.tcl script, and the offset values are subtracted from the corresponding sampling schedule. It can take as many as 3 space- or comma-separated numbers for a 4D data set. The default value is zero. If the indices in the sampling schedule for a particular dimension are not zero-based, then the offset should be 1 for that dimension. This option is particularly useful for an indirect dimension acquired with one point delay and therefore with 360° first order phase correction. This is the case in the 3D HNCO example presented below in Section 6.3. The C' dimension (z-dimension in fid.com, but becomes y-dimension during the SMILE reconstruction) was acquired with a delay equal to one full increment. The offset of "0 -1" (or "0, -1") actually shifts the y-dimension (i.e. the C' dimension) sampling schedule forward by 1. As a result, the point with t_2=0 is treated as an unsampled point, and the first actually sampled point becomes the second point, and so forth. This is equivalent to the time-shifting prior to the Fourier transform and eliminates the required 360° first order phase correction and the concomitant constant baseline offset in the C' dimension.

5.14 -report

Although not displayed in the help page of the beta SMILE version, this option (default: 0) can be set to 1 for saving all critical reconstruction parameters in the log file "smile.log". In addition, this log file includes two RMS values for each cross section along the direct dimension: one for the input data before the first iteration and the other for the residual spectrum after the last iteration. The difference between the two RMS values is related to the quality of the reconstruction. In an ideal case, the residual RMS after the reconstruction can approach the thermal (and t_1) noise level, while cross sections with a high number of strong peaks may have slightly (20-30%) elevated residual RMS values. Note that this file will be overwritten if users run the SMILE job again.

5.15 Other SMILE options used during the development

Several other SMILE options that are not discussed here for the time being are included in the SMILE help page (see Section 2 above or from the "nmrPipe -fn SMILE -help" command). These were used primarily for development purposes and may be removed in future versions. Users generally do not need those to optimize a reconstruction.

6. SMILE examples

To demonstrate how SMILE is used for a good reconstruction, users can download 5 examples from:

http://spin.niddk.nih.gov/bax/software/SMILE

Each example directory includes all the original Topspin 2.1 files, the expansion and conversion script fid.com as well as the processing script smile.com. To avoid overwriting the original files, users are recommended to rename all script files and output directories.

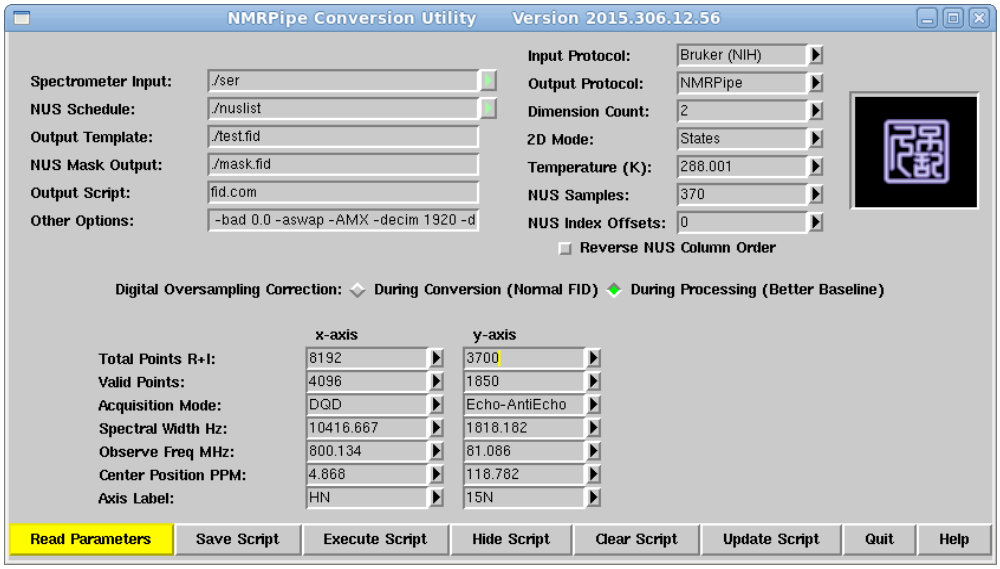
6.1 2D TROSY 20% NUS reconstruction

This is a NUS data set artificially made by dropping 80% of fully sampled points in ser.orig, and the ser file randomly retains the remaining 20% indirect points according to the sampling schedule in nuslist. The spectrum fullSample.ft2 was obtained by processing the fully sampled ser.orig data and provides a benchmark to evaluate how good the reconstruction is. The other spectrum noSmile.ft2 was obtained from the 20% sparsely sampled ser without SMILE reconstruction, and clearly shows strong NUS sampling artifacts.

The first step to process the NUS data is to set up an NMRPipe conversion script (i.e. fid.com) that sorts the ser file according to the schedule in nuslist and fills all the unsampled points with zeros. The "bruker" command in the latest NMRPipe

package has been updated to facilitate this set-up process. Although "bruker" can automatically detect the nuslist file and enter its NUS mode, a command line with explicit NUS related options (i.e. "bruker -nus -nouseMask") is recommended. Here the option "-nus" tells "bruker" to enter the NUS mode, while "-nouseMask" disables the generation of a mask file that is as large as the expanded time-domain data but not needed for SMILE. If the sampling schedule is not in the nuslist file, users can pass the filename to the command using "-sample filename" option. Enter "bruker -help" on a terminal for more details about additional command line options. It is worth pointing out that a conversion script can be set up in the same way for Varian NUS data by using "varian -nus -nouseMask". See the help page printed by "varian –help" for more information.

The "bruker -nus -nouseMask" brings up the following graphics user interface that should be very similar to the one for a conventional data set, but contains additional fields for users to provide the NUS related information:



User should make sure the correct NUS sampling list file is selected for the "NUS Schedule" field. For the "NUS Samples", the default value is the length of the sampling list (i.e. number of elements, pairs, or triples in the sampling schedule for 2D, 3D, and 4D, respectively), and this value is used to set the "-sampleCount" option for nusExpand.tcl during the expansion. If a NUS experiment stops before the entire sampling list is read or if users just want to further down sample the already collected data for whatever reasons, this field can be manually changed to a smaller value. Note that any change made to "NUS Sample" makes the "Read Parameters" button highlighted, prompting users to click it again for updating the "Valid Points" and "Total Points R+I". This is needed because cutting the sampling list down may change the largest sampling value, which requires the "Valid Points" to be set accordingly for a correct data conversion. In this example, all 370 points defined in the nuslist file were sampled, and the largest value in the list is 1849. Because the

sampling index is zero based, i.e. from 0 to 1849, the expanded data (default output as "ser_full") by nusExpand.tcl consists of 1850 complex pairs.

The "NUS Index Offsets" should be set to 0 because the ^{15}N sampling list nuslist is zero based and there is no time-shifting needed for this example to remove any first order phase correction. Note that the digital oversampling correction preferentially is made (automatically by NMRPipe) during processing instead of during conversion (using the -AMX option), which sometimes can improve the spectral baseline.

Since the data is sorted and expanded prior to the conversion, Echo-AntiEcho can be used as the "Acquisition Mode" for the y-axis. For Bruker users who collect data in a standard way, it is likely that clicking the "Read Parameters" button can set the "Observe Freq MHz", "Center Position PPM" and "Axis Label" entries correctly. Make sure that the "Center Position PPM" should be set to "H2O" for the x-axis prior to each click if the carrier is on the water resonance. Otherwise the center positions may need to be calculated and entered manually for each dimension. Note that all the example data were collected with some unusual Topspin set-up and users may need to manually update the values in several fields because not all the information required by "bruker" is available from the Bruker acquisition files. With all entries correctly set and updated, clicking the "Save Script" button generates the following fid.com script:

```
#!/bin/csh

nusExpand.tcl -mode bruker -sampleCount 370 -off 0 \
 -in ./ser -out ./ser_full -sample ./nuslist

bruk2pipe -in ./ser_full \
  -bad 0.0 -aswap -AMX -decim 1920 -dspfvs 20 -grpdly 67.9841918945312  \
  -xN              8192  -yN              3700  \
  -xT              4096  -yT              1850  \
  -xMODE            DQD  -yMODE  Echo-AntiEcho  \
  -xSW        10416.667  -ySW         1818.182  \
  -xOBS         800.134  -yOBS          81.086  \
  -xCAR           4.868  -yCAR         118.782  \
  -xLAB              HN  -yLAB             15N  \
  -ndim               2  -aq2D          States  \
  -out ./test.fid -verb -ov
```

The above script first sorts and expands the ser file into ser_full, followed by a conventional data conversion from ser_full to test.fid in the NMRPipe format. The smile reconstruction script below starts with a conventional processing of the direct dimension, followed by the matrix transposition such that the direct dimension is stored in the y-dimension while the indirect NUS dimension becomes the x-axis, after which "nmrPipe -fn SMILE" can begin the reconstruction:

```
#!/bin/csh

nmrPipe -in test.fid                                    \
| nmrPipe  -fn POLY -time                               \
| nmrPipe  -fn GMB -lb -4 -gb 0.8 -c 1.0                \
| nmrPipe  -fn ZF -zf 2 -auto                           \
| nmrPipe  -fn FT                                       \
| nmrPipe  -fn PS -p0 -24 -p1 0 -di                     \
| nmrPipe  -fn POLY -auto -ord 2 -x1 10ppm -xn 6ppm     \
| nmrPipe  -fn EXT -x1 8.8ppm -xn 7.8ppm -sw -round 2   \
```

```
| nmrPipe  -fn TP                                          \
| nmrPipe  -fn SMILE -nDim 2 -sample nuslist -maxIter 500 \
           -nThread 4 -nSigma 4 -xP0 90 -xP1 0 -report 1  \
| nmrPipe  -fn ZF -zf 2 -auto                              \
| nmrPipe  -fn FT                                          \
| nmrPipe  -fn PS -p0  90  -p1  0  -di                    \
| nmrPipe  -fn TP                                          \
   -verb -ov -out smile.ft2
```

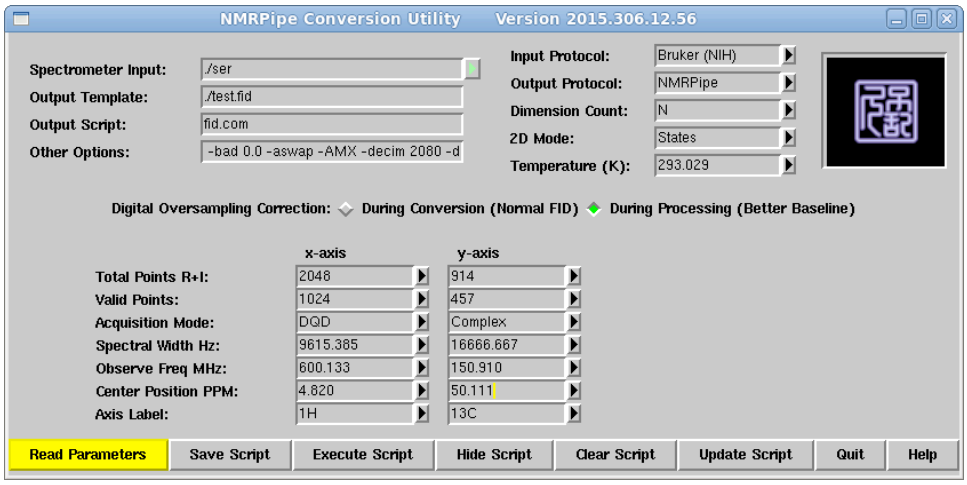
Note that "nmrPipe -fn POLY -auto -ord 2 -x1 10ppm -xn 6ppm" is applied after the phase correction is made, but before the narrow spectral region of interest is extracted. This may help improve the baseline along the direct dimension, but may slightly increase the noise level in the final data set. Users should take this step with caution as it may actually distort the baseline if the resonance is more dispersed such that not enough baseline segments are available. In practice, users can check if such a baseline correction is beneficial by examining the spectrum without SMILE reconstruction (or by setting -maxIter to 0). It also must be mentioned that the "-round 2" option must be included for the "nmrPipe -fn EXT" function to ensure the direct dimension to have an even number of points. Otherwise, SMILE quits with an error message in case the direct dimension gets an odd number of points.

Most command line options used here are self explanatory. For 2D, "-nSigma 4" can be used for a slightly cleaner reconstruction. "-report 1" instructs SMILE to save some critical parameters in "smile.log" (see Section 5.14 for more details). Note that there is no apodization function specified for the ^{15}N dimension. SMILE applies the default function (-xApod SP –xQ1 0.5 –xQ2 0.98 –xQ3 1) to apodize the data. The data remains to be apodized with the first point properly scaled and therefore there is no need to apply any window function prior to the conventional FT step after the reconstruction completes.

The final reconstructed spectrum is in smile.ft2, and users can compare this with fullSample.ft2 and noSmile.ft2.

6.2 SMILE as an alternative to LP for extending the constant-time acquisition from 28 ms to 56 ms in a fully sampled CT-[^{13}C-^{1}H]-HSQC experiment

This example shows how SMILE can be used as an alternative method to linear prediction for extending the acquisition time of a fully sampled indirect dimension by treating it as a special NUS data set. The conversion script is first created by using the "bruker -nonus" command. Although not typically required for a fully sampled data set, the "-nonus" option explicitly instructs "bruker" to enter the conventional conversion mode. Without this option, "bruker" automatically starts in its NUS conversion mode when the file "nuslist" (created by nusZF.com, see below) exists.



Note again that the "Center Position PPM" for the x-axis should be set to "H2O" prior to each clicking of the "Read Parameters" button if the carrier was placed on water during the experiment. Otherwise, users may need to manually calculate and update the carrier positions in both dimensions. Once all the entries are updated and manually corrected if necessary, click the "Save Script" button to generate the following conventional fid.com script without the nusExpand.tcl line:

```
#!/bin/csh

bruk2pipe -in ./ser \
  -bad 0.0 -aswap -AMX -decim 2080 -dspfvs 20 -grpdly 67.9842071533203  \
  -xN              2048  -yN               914  \
  -xT              1024  -yT               457  \
  -xMODE            DQD  -yMODE         Complex  \
  -xSW         9615.385  -ySW        16666.667  \
  -xOBS         600.133  -yOBS         150.910  \
  -xCAR           4.820  -yCAR          50.111  \
  -xLAB              1H  -yLAB             13C  \
  -ndim               2  -aq2D          States  \
  -out ./test.fid -verb -ov
```

The NMRPipe script nusZF.com (provided in the new release by Frank Delaglio) starts from the fully sampled test.fid and creates a new data set (test_ext.fid) with zeros padded in the indirect dimension. By setting "-yZFARG" to "zf=1", the program increases the CT acquisition by exactly one fold, i.e. from 28 ms to 56 ms. The "-schedule" option allows user to provide a file name for the artificial NUS sampling schedule to be created during the extension. Use "nusZF.com -help" for more command line options.

```
nusZF.com -in test.fid -out test_ext.fid \
          -schedule nuslist -mask None   \
          -yZFARG zf=1
```

The extended data can be considered as 50% sparsely sampled and can be reconstructed using SMILE in a similar way as described above for the 2D TROSY example (Section 6.1) by using the following script:

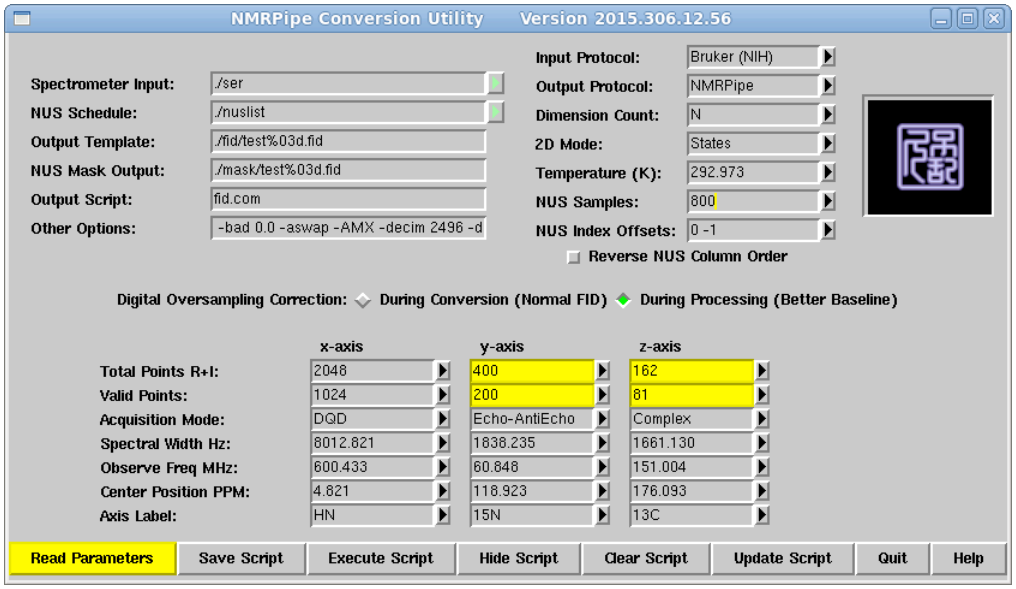
```
#!/bin/csh

nmrPipe -in test_ext.fid \
| nmrPipe  -fn POLY -time                                  \
| nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 1.0         \
| nmrPipe  -fn ZF -zf 2 -auto                              \
| nmrPipe  -fn FT                                          \
| nmrPipe  -fn PS -p0 47.9 -p1 37.2 -di -verb              \
| nmrPipe  -fn EXT -x1 6.5ppm -xn -2.0ppm -sw              \
| nmrPipe  -fn TP                                          \
| nmrPipe  -fn SMILE -nDim 2 -sample nuslist -nThread 4    \
            -nSigma 4 -maxIter 1000 -report 1              \
            -xApod SP -xQ1 0.5 -xQ2 0.995 -xQ3 1           \
| nmrPipe  -fn ZF -zf 2 -auto                              \
| nmrPipe  -fn FT                                          \
| nmrPipe  -fn PS -p0 0 -p1 0 -di -verb                    \
| nmrPipe  -fn TP                                          \
| nmrPipe  -fn POLY -auto                                  \
 -ov -out smile.ft2
```

6.3 3D HNCO 5% NUS reconstruction

This 3D HNCO serves as an example for processing the NUS data collected with the experiment terminated before its completion. As for the 2D, we start with the command "bruker -nus -nouseMask" to construct the expansion and conversion script. Although the nuslist consists of 8000 (^{15}N, ^{13}C) sampling pairs, only 10% of the total NUS data points were sampled. As a result, the "NUS Samples" should be manually updated to 800 from the default value 8000. Note that the change in the "NUS Samples" makes the "Read Parameters" as well as the "Valid Points" and "Total Points R+I" entries highlighted, reminding users of clicking the button again to update the number of points that may or may not become smaller. Remember to manually set the "Center Position PPM" to "H2O" before each click in order for all carrier positions to be correctly calculated, unless the carrier was placed elsewhere.

At this fid.com conversion step, the first column in the sampling schedule corresponds to the y-axis, while the second column to the z-axis, unless the "Reverse NUS Column Order" is checked. The ^{13}C dimension (i.e. along the z-axis) was originally recorded with one full point delay for 80 complex points in total indexed from 0 to 79. By setting the second offset to -1 in the "NUS Index Offset" in the figure below, we shift the second column in the sampling schedule (i.e. corresponding to the ^{13}C sampling list) from an index range of 0-79 to 1-80. This is because the offset is always subtracted by nusExpand.tcl (and SMILE as well) from the index in the sampling list, thus 0 - (-1) = 1, ..., and 79 - (-1) = 80. Because nusExpand.tcl always interprets any offset-corrected sampling schedule as zero-based, shifting the index forward by 1 is equivalent to treating the first t_2=0 data point as an additional unsampled point, thereby effectively changing the ^{13}C acquisition from one point delay to no delay. This manipulation eliminates the otherwise required 360° first order phase correction as well as the baseline offset in the corresponding dimension. Note that the "Valid Points" for the z-axis must be manually changed from 80 to 81 and the "Total Points R+I" must be updated from 160 to 162 to accommodate the addition of the unsampled point. Note that the -1 offset is applied only internally by nusExpand.tcl when sorting and zero filling the

NUS data. The file (i.e. nuslist) containing the sampling schedule remains unchanged during the conversion. No new offset-corrected sampling schedule text file is created either. As a result, the same offset must be applied during SMILE reconstruction, and the SMILE "-off" option is designed to do just that.



The above setting generates the following fid.com script:

```
#!/bin/csh

nusExpand.tcl -mode bruker -sampleCount 800 -off 0 -1 \
 -in ./ser -out ./ser_full -sample ./nuslist

bruk2pipe -in ./ser_full \
  -bad 0.0 -aswap -AMX -decim 2496 -dspfvs 20 -grpdly 67.9842376708984 \
  -xN              2048  -yN               400  -zN               162  \
  -xT              1024  -yT               200  -zT                81  \
  -xMODE            DQD  -yMODE  Echo-AntiEcho  -zMODE        Complex  \
  -xSW         8012.821  -ySW         1838.235  -zSW         1661.130  \
  -xOBS         600.433  -yOBS          60.848  -zOBS         151.004  \
  -xCAR           4.821  -yCAR         118.923  -zCAR         176.093  \
  -xLAB              HN  -yLAB             15N  -zLAB             13C  \
  -ndim               3  -aq2D          States                         \
  -out ./fid/test%03d.fid -verb -ov
```

After running the above script to complete the expansion and conversion, the direct dimension is then processed and stored in the z-dimension:

```
#!/bin/csh

xyz2pipe -in ./fid/test%03d.fid -x                    \
| nmrPipe  -fn POLY -time                             \
| nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 0.5    \
| nmrPipe  -fn ZF -zf 2 -auto                         \
| nmrPipe  -fn FT                                     \
| nmrPipe  -fn POLY -auto -ord 2 -x1 11ppm -xn 6ppm   \
| nmrPipe  -fn EXT -x1 10ppm -xn 6.4ppm -sw -round 2  \
| nmrPipe  -fn PS -p0 -58 -p1 0.0 -di                 \
| pipe2xyz -out ft1/test%04d.ft1 -z
```

Note that a baseline correction is applied using a wider range (6-11 ppm) than the

extracted region (6.4-10ppm), which may improve the baseline. Also the "-round 2" option is used to ensure an even number of points in the direct dimension after being extracted. The data is output to ft1/test%04d.ft1 via pipe2xyz along the -z option, which permutes the original xyz-axes to yzx, thereby transposing the current axis (i.e. the direct dimension) from the x- to z-axis, but keeping the relative axis order of the two indirect dimensions (i.e. the original y- and z-axes becoming the x- and y-axes, respectively). The same axis order can be obtained through ZTP by first swapping the x- with z-axes followed by TP to rotate the y-axis to the first dimension. Regardless, users should use the command "showhdr ft1/test%04d.ft1" to check the axis order:

```
FILE: ft1/test0001.ft1 DIM: 3 QUAD: Complex 2DMODE: States Not Transposed
BYTES: 261248 PRED: 261248 MIN: 0 MAX: 0 VALID: 0
ORDER: 1 3 2 PIPE: 0 CUBE: 0 FILES: 1106 200x162x2 2D Series

               X-Axis        Y-Axis        Z-Axis

DATA SIZE:            200           162          1106
APOD SIZE:            200            81           276
SW Hz:        1838.234985   1661.130005   2163.618164
OBS MHz:        60.848000    151.003998    600.432983
ORIG Hz:      6326.361328  25770.585938   3843.471191
DOMAIN:              Time          Time          Freq
MODE:             Complex       Complex          Real
NAME:                 15N           13C            HN
```

Note that SMILE always assigns the first column in the sampling schedule to the x-axis, and the second column to the y-axis. If this is not the case, users must either transpose the data or swap the sampling list such that the axis order in the ft1/test%04d.ft1 and nuslist is identical. Otherwise, SMILE will encounter an inconsistency between the sampling list and the data size of each indirect dimension (unless any two indirect dimensions happen to have the same size), and quits after an error is reported.

```
xyz2pipe -in ft1/test%04d.ft1 -x                              \
| nmrPipe  -fn SMILE -nDim 3 -sample nuslist -nThread 32      \
           -sampleCount 800 -nSigma 5 -off 0 -1 -report 1     \
           -xCT 43                                            \
| pipe2xyz -out ft1/rc%04d.ft1 -x
```

The command line above performs a 3D SMILE reconstruction. Note that -sampleCount must be specified and be consistent with the value in the nusExpand.tcl script, unless users truncate the sampling schedule and keep the top 800 lines only. The "-off" option also needs to be set to the same values used for nusExpand.tcl, and the -1 offset for the second column effectively shifts the ^{13}C sampling list forward by 1 as discussed above, but the shifting only occurs internally within SMILE, which does not change the nuslist file. Like nusExpand.tcl, the shifted sampling list is considered as zero-based by SMILE.

The ^{15}N dimension was acquired using the so-called mixed time evolution approach (see Ying et al., J. Biomol. NMR, 2007, 37, 195-204). The first 44 complex pairs were recorded in a constant time manner, while the additional points through real time increments. As the sampling list is zero based, the "-xCT 43" instructs SMILE that exactly 44 complex pairs out of 200 belong to the constant time evolution.

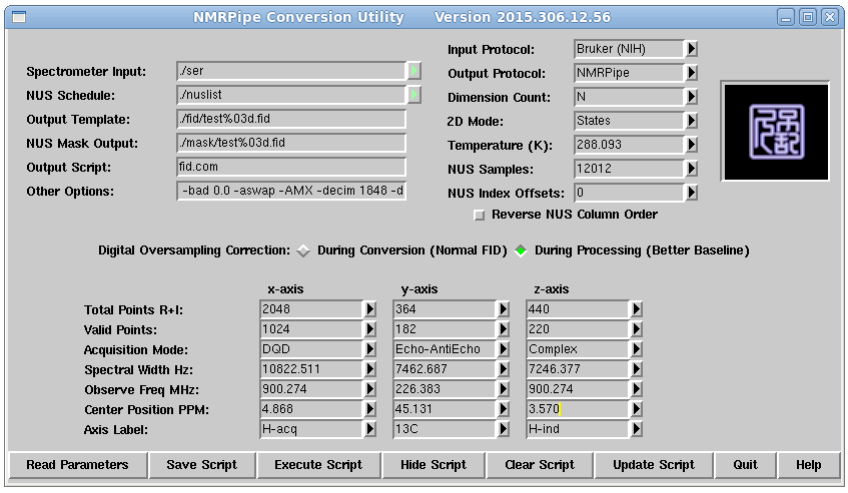
Although the SMILE function can be connected via a unix pipe to the following conventional processing of the two indirect dimensions, we recommend to first output the results to ft1/rc%04d.ft1 files. This allows users to play with the subsequent processing without repeating the time-consuming reconstruction step. Note that no window function is applied in the following conventional script because the data was already apodized during the above SMILE reconstruction using the default parameters (-Apod SP -Q1 0.5 -Q2 0.98 -Q3 1 for both ^{15}N and ^{13}C).

```
xyz2pipe -in ft1/rc%04d.ft1 -x                    \
| nmrPipe  -fn ZF -zf 1 -auto                     \
| nmrPipe  -fn FT                                 \
| nmrPipe  -fn PS -p0 0 -p1 0 -di                 \
| nmrPipe  -fn TP                                 \
| nmrPipe  -fn ZF -zf 2 -auto                     \
| nmrPipe  -fn FT                                 \
| nmrPipe  -fn PS -p0 0 -p1 0 -di                 \
| nmrPipe  -fn TP                                 \
| nmrPipe  -fn ZTP                                \
| pipe2xyz -out ft/test%04d.ft3 -x

proj3D.tcl -in ft/test%04d.ft3
```

6.4 3D ^{13}C NOESY-HSQC 30% NUS reconstruction

This example demonstrates that SMILE works in the presence of strong t_1 noise ridges. It also shows how the indirect dimensions are apodized differently with GM and SP during the reconstruction, but during the subsequent conventional processing the GM window is inverted and an SP function is applied instead.

The expansion and conversion script is first constructed by entering "bruker -nus -nouseMask". Make sure the "NUS Schedule" and "NUS Samples" fields are correctly set, and the "Center Position PPM" has "H2O" before each click of the "Read Parameters" button. Manually update the other entries if necessary before the script is saved. The carrier frequency in the indirect ^{1}H dimension was shifted to 3.4 ppm in the pulse sequence. But this is relative to the O1P of 4.698. As a result, the correct position should be 3.400 + 4.868 – 4.698 = 3.570 ppm.



The above “bruker” settings lead to the following fid.com conversion script:

```
#!/bin/csh

nusExpand.tcl -mode bruker -sampleCount 12012 -off 0 \
 -in ./ser -out ./ser_full -sample ./nuslist

bruk2pipe -in ./ser_full \
  -bad 0.0 -aswap -AMX -decim 1848 -dspfvs 20 -grpdly 67.9869537353516  \
  -xN              2048  -yN               364  -zN               440  \
  -xT              1024  -yT               182  -zT               220  \
  -xMODE            DQD  -yMODE  Echo-AntiEcho  -zMODE        Complex  \
  -xSW        10822.511  -ySW         7462.687  -zSW         7246.377  \
  -xOBS         900.274  -yOBS         226.383  -zOBS         900.274  \
  -xCAR           4.868  -yCAR          45.131  -zCAR           3.570  \
  -xLAB           H-acq  -yLAB             13C  -zLAB           H-ind  \
  -ndim               3  -aq2D          States                         \
  -out ./fid/test%03d.fid -verb -ov
```

After running fid.com, the direct dimension must be processed and transposed to the z-axis using the pipes as follows:

```
xyz2pipe -in fid/test%03d.fid                          \
| nmrPipe  -fn POLY -time                              \
| nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 0.5     \
| nmrPipe  -fn ZF -zf 1 -auto                          \
| nmrPipe  -fn FT                                      \
| nmrPipe  -fn EXT -x1 6ppm -xn -0.5ppm -sw -round 2   \
| nmrPipe  -fn PS -p0 -9.0 -p1 0 -di                   \
| pipe2xyz -ov -out ft1/test%04d.ft1 -z
```

Note that in this example, there is no POLY baseline correction applied before the region of interest was extracted. For a NOESY spectrum like this, there are not enough baseline segments in each 1D slice to make a good correction. Also the direct dimension is zero filled once, which already yields 1110 points and a sufficiently high digital resolution (ca 1.3 Hz). The SMILE reconstruction can now begin using the following script:

```
xyz2pipe -in ft1/test%04d.ft1 -x -verb                 \
| nmrPipe -fn SMILE -nDim 3 -maxIter 600 -nSigma 6     \
            -sample nuslist -nThread 64 -thresh 0.95   \
            -xApod GM -xQ1 0   -xQ2 50    -xQ3 0       \
            -yApod SP -yQ1 0.5 -yQ2 0.98  -yQ3 1       \
            -xzfSize 728 -xP0 90 -xP1   0 -xCT 182     \
            -yzfSize 880 -yP0 90 -yP1 180 -report 1    \
| pipe2xyz -out ft1.GM/rc%04d.ft1 -x
```

The indirect dimensions are apodized differently with GM and SP for x- and y-axis, respectively. This is done just to show the syntax of using different windows for SMILE reconstruction. There is no particular advantage of applying GM in this case, which makes the CT ^{13}C peaks broader. The following script shows how the GM window in the ^{13}C dimension is inverted through the “-inv” option of the GM function. Note the “-hdr” option instructs GM to read the Q values of the GM window from the header. This can be done because SMILE saves the apodization parameters of each indirect dimension in the header. However, the first-point scaling factor for the “-c” option is not saved in the header and therefore the scaling was not inverted. As a result, no scaling should be repeated when the new SP window is applied (i.e. -c set to 1.0 instead of 0.5). For the y-axis (i.e. the ^{1}H dimension), no SP function is

applied as the data is already apodized by SMILE unless a different window is needed.

```
xyz2pipe -in ft1.GM/rc%04d.ft1 -x                    \
| nmrPipe  -fn GM -hdr -inv                          \
| nmrPipe  -fn SP -off 0.5 -end 0.995 -pow 1 -c 1.0  \
| nmrPipe  -fn ZF -zf 2 -auto                        \
| nmrPipe  -fn FT                                    \
| nmrPipe  -fn PS -p0 90 -p1 0 -di                   \
| nmrPipe  -fn TP                                    \
| nmrPipe  -fn ZF -zf 2 -auto                        \
| nmrPipe  -fn FT                                    \
| nmrPipe  -fn PS -p0 90 -p1 180 -di                 \
| nmrPipe  -fn TP                                    \
| nmrPipe  -fn ZTP                                   \
| nmrPipe  -fn POLY -auto -ord 2                     \
| pipe2xyz -out ft.GM/test%04d.ft3 -x

proj3D.tcl -in ft.GM/test%04d.ft3
```

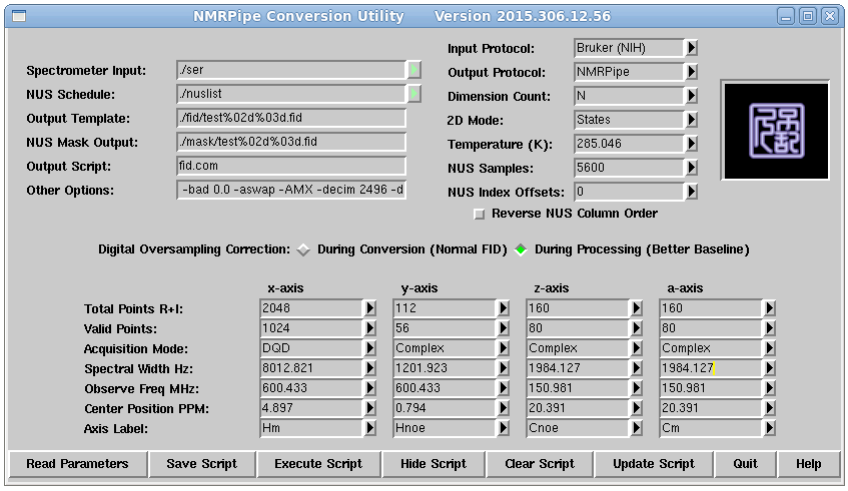
For comparison, the reconstruction can be done using a cosine window for both indirect dimensions using the processing scripts below. The settings of "-Apod", "-Q1", "-Q3" with no axis designation apply to both indirect dimensions, while "-xQ2 0.995" and "-yQ2 0.98" cut the sine windows at 179.1° for the CT ^{13}C dimension and 172.4° for the regular ^{1}H dimension, respectively.

```
xyz2pipe -in ft1/test%04d.ft1 -x -verb                \
| nmrPipe -fn SMILE -nDim 3 -maxIter 600 -nSigma 6     \
            -sample nuslist -nThread 32 -thresh 0.95   \
            -Apod SP -Q1 0.5 -xQ2 0.995 -yQ2 0.98 -Q3 1 \
            -xzfSize 728 -xP0 90 -xP1   0 -xCT 182     \
            -yzfSize 880 -yP0 90 -yP1 180 -report 1    \
| pipe2xyz -out ft1.SP/rc%04d.ft1 -x

xyz2pipe -in ft1.SP/rc%04d.ft1 -x                     \
| nmrPipe  -fn ZF -zf 2 -auto                         \
| nmrPipe  -fn FT                                     \
| nmrPipe  -fn PS -p0 90 -p1 0 -di                    \
| nmrPipe  -fn TP                                     \
| nmrPipe  -fn ZF -zf 2 -auto                         \
| nmrPipe  -fn FT                                     \
| nmrPipe  -fn PS -p0 90 -p1 180 -di                  \
| nmrPipe  -fn TP                                     \
| nmrPipe  -fn ZTP                                    \
| nmrPipe  -fn POLY -auto -ord 2                      \
| pipe2xyz -out ft.SP/test%04d.ft3 -x

proj3D.tcl -in ft.SP/test%04d.ft3
```

6.5 4D methyl HMQC-NOESY-HMQC 1.56% NUS reconstruction

Enter the command "bruker -nus -nouseMask" to set up the expansion and conversion scripts. Click the "Read Parameters" button, and make sure the "NUS Schedule" is set to nuslist and the "NUS Samples" to 5600 (equal to the number of triples in the nuslist for the completed experiment). The "Acquisition Mode" for all indirect dimensions should be set to "Complex", and the "Observe Freq MHz" should be set to the values as in the figure below. Make sure the "Spectral Width Hz" is updated manually, if necessary. The "Center Position PPM" should be set to "H2O" for the x-axis **before** clicking the "Read Parameters" button each time, which then allows users to choose a correctly calculated carrier frequency for the other

dimensions. If the carrier position along the x-axis is set to a number in ppm, clicking the "Read Parameters" button changes the position to 0 ppm, which would then require users to manually enter the center positions for all the dimensions. Note that for the carrier position along the y-axis, it is set to 0.6 ppm in the pulse sequence, but that is the position relative to the O1P at 4.703 ppm. At 285K, water resonates at 4.897 ppm. As a result, the correct center for the y-axis is 4.897 – 4.703 + 0.6 = 0.794 ppm. For the "Axis Label" along each axis, users can use a unique string that helps identify each dimension. To make sure a manual change is updated in the script before it is saved, users must hit the enter key after the change is made.



The above setting yields the following fid.com script:

```
#!/bin/csh

nusExpand.tcl -mode bruker -sampleCount 5600 -off 0 \
 -in ./ser -out ./ser_full -sample ./nuslist

bruk2pipe -in ./ser_full \
  -bad 0.0 -aswap -AMX -decim 2496 -dspfvs 20 -grpdly 67.9842376708984  \
  -xN              2048  -yN               112  -zN               160  -aN               160  \
  -xT              1024  -yT                56  -zT                80  -aT                80  \
  -xMODE            DQD  -yMODE         Complex  -zMODE         Complex  -aMODE         Complex  \
  -xSW         8012.821  -ySW          1201.923  -zSW          1984.127  -aSW          1984.127  \
  -xOBS         600.433  -yOBS          600.433  -zOBS          150.981  -aOBS          150.981  \
  -xCAR           4.897  -yCAR            0.794  -zCAR           20.392  -aCAR           20.392  \
  -xLAB              Hm  -yLAB             Hnoe  -zLAB             Cnoe  -aLAB              Cm  \
  -ndim               4  -aq2D           States                                               \
| pipe2xyz -x -out ./fid/test%03d%03d.fid -verb -ov
```

After running the fid.com script, users can begin the conventional processing of the direct dimension. As pointed out in the other examples, a baseline correction is applied to a wider range than the spectral region of interest, before the region is extracted. The "-round 2" option ensures the direct dimension to have an even number of points. Finally, the processed x-dimension is permutated to the a-axis, while the original y-, z-, and a-axis become x-, y-, and z-axis, respectively. The relative axis order of the indirect dimensions must be preserved and they must correspond to the three columns in the NUS sampling schedule. Otherwise, either the columns in the sampling list need to be swapped or the data matrix must be

rearranged. Regardless, the direct dimension must be processed with its imaginaries discarded, and must be stored along the a-axis for a 4D data set.

```
#!/bin/csh

xyz2pipe -in fid/test%03d%03d.fid -x                    \
| nmrPipe  -fn POLY -time                               \
| nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 0.5      \
| nmrPipe  -fn ZF -zf 2 -auto                           \
| nmrPipe  -fn FT                                       \
| nmrPipe  -fn POLY -auto -ord 2 -x1 3ppm -xn -1ppm     \
| nmrPipe  -fn EXT -x1 1.4ppm -xn -0.7ppm -sw -round 2  \
| nmrPipe  -fn PS -p0 122 -p1 0 -di                     \
| pipe2xyz -ov -out ft1/test%03d%03d.ft1 -a
```

The data is now ready for the 4D reconstruction using the script below. Note that the reconstruction is allowed to access 490 GB of memory via "-maxMem 490". As discussed above, around 17 GB space is minimally required. Also a two-fold zero fill without being rounded to the next power of 2 is used via the "-yzfSize" and "-zzfSize" options for the two ^{13}C dimensions to keep the total data size smaller. The "-xNeg" and "-yNeg" options instruct SMILE to negate the imaginary during the processing, as required by the conventional FT function during the subsequent processing.

```
xyz2pipe -in ft1/test%03d%03d.ft1 -x                    \
| nmrPipe -fn SMILE -nDim 4 -maxIter 200 -nSigma 6      \
            -sample nuslist -nThread 60                 \
            -maxMem 490 -report 1                       \
            -xzf       2 -xP0 0.0 -xP1 0.0 -xNeg        \
            -yzfSize 320 -yP0 0.0 -yP1 0.0 -yNeg        \
            -zzfSize 320 -zP0 90  -zP1 180              \
| pipe2xyz -out ft1/rc%03d%03d.ft1 -x

xyz2pipe -in ft1/rc%03d%03d.ft1 -x                      \
| nmrPipe  -fn ZF -zf 2 -auto                           \
| nmrPipe  -fn FT -neg                                  \
| nmrPipe  -fn PS -p0 0 -p1 0 -di                       \
| nmrPipe  -fn TP                                       \
| nmrPipe  -fn ZF -zf 1 -auto                           \
| nmrPipe  -fn FT -neg                                  \
| nmrPipe  -fn PS -p0 0 -p1 0 -di                       \
| pipe2xyz -out ft/test%03d%03d.ft3 -y

xyz2pipe -in ft/test%03d%03d.ft3 -z                     \
| nmrPipe  -fn ZF -zf 1 -auto                           \
| nmrPipe  -fn FT                                       \
| nmrPipe  -fn PS -p0 90 -p1 180 -di                    \
| pipe2xyz -out ft/test%03d%03d.ft4 -z

proj4D.tcl -in ft/test%03d%03d.ft4
```

7. Contact and reference

Contact Jinfa Ying at jinfaying@niddk.nih.gov for any bugs, comments and suggestions. Your feedbacks are much appreciated. Cite the work by referencing to Ying et al., J. Biomol. NMR, in preparation.

8. Acknowledgements

This manual is written by Jinfa Ying and Ad Bax. We thank Dennis Torchia for his contribution to the initial formulation of the SMILE algorithm and for his many helpful discussions during the development. We are also grateful to Frank Delaglio for providing his source codes of NMRPipe, for answering many questions from us, and for his development of routines that facilitate the handling of NUS data prior to SMILE reconstruction. We thank Alex Maltsev and Yang Shen for their helpful discussions in coding the algorithm. Jung Ho Lee, Nik Sgourakis, Julien Roche, Fang Li, Yawen Bai (NCI), and Hanqiao Feng (NCI) are thanked for providing the samples and data used in this work.